Flights
Airlines
Airports

Using Databricks, Spark, and Snowflake for flight data ETL and analysis

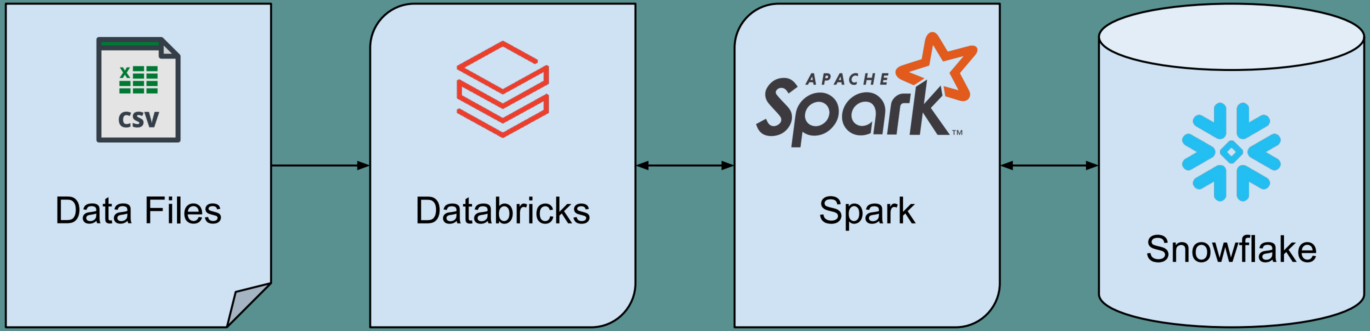
Tools

databricks

snowflake

Spark

Data Flow



Data Files

Databricks

Spark

Snowflake

Data

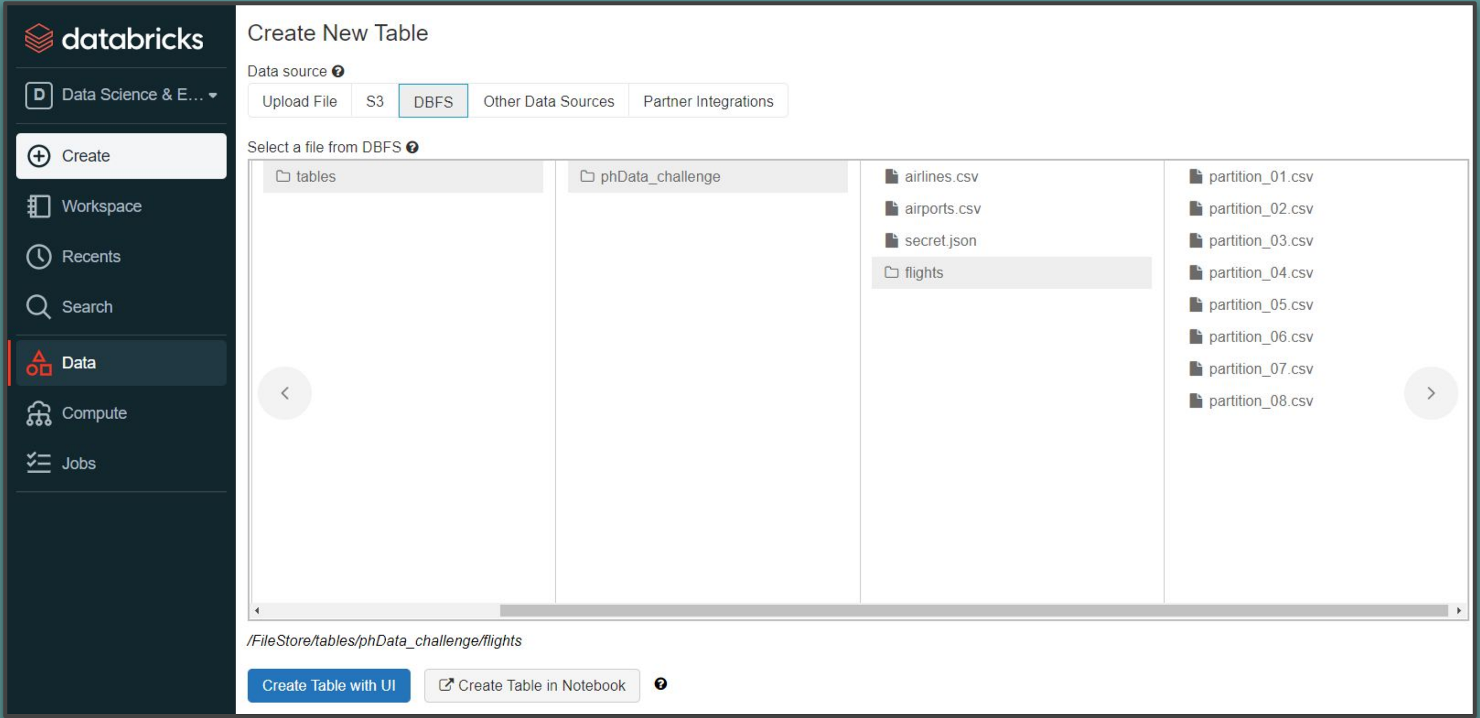
CSV Files

- 14 Airlines, 1 file
- 322 Airports, 1 file
- 3,920,766 Flights, 8 files

3,920,766 Flights!

Databricks

DBFS contains CSV Files



Create New Table

Data source

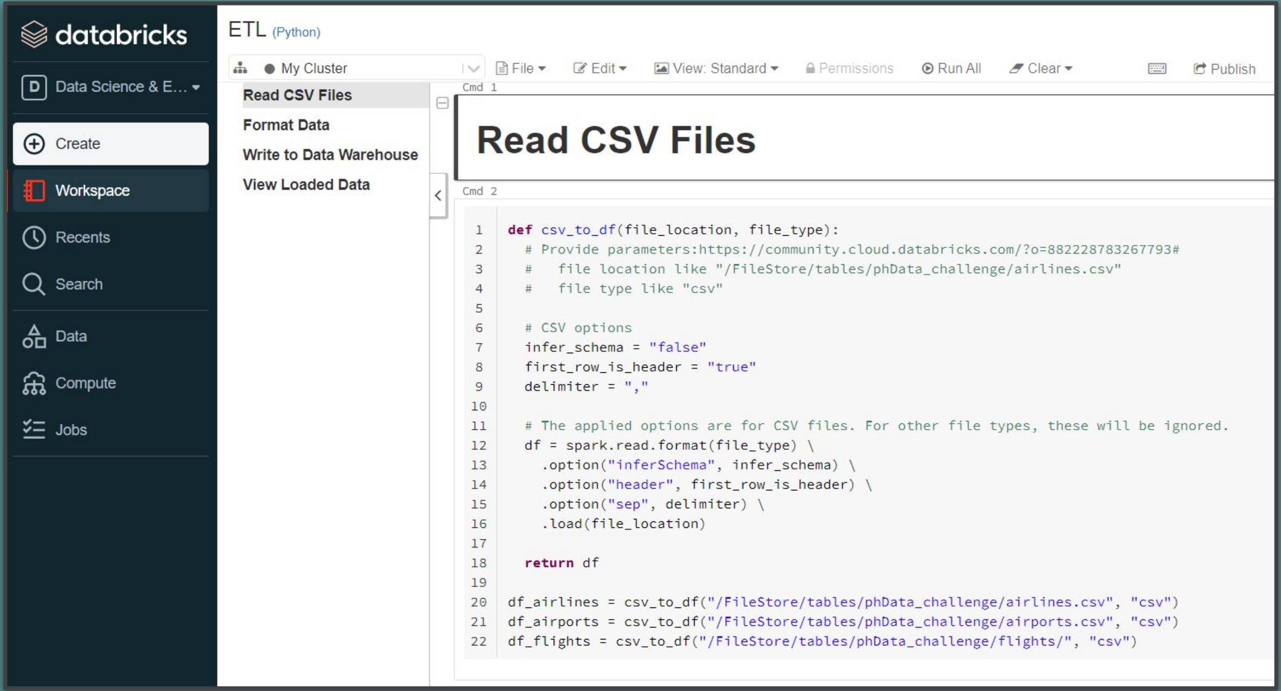
Select a file from DBFS

- tables
- phData_challenge
- airlines.csv
- airports.csv
- secret.json
- flights
- partition_01.csv
- partition_02.csv
- partition_03.csv
- partition_04.csv
- partition_05.csv
- partition_06.csv
- partition_07.csv
- partition_08.csv

/FileStore/tables/phData_challenge/flights

Spark

Extract, Transform, Load (ETL)

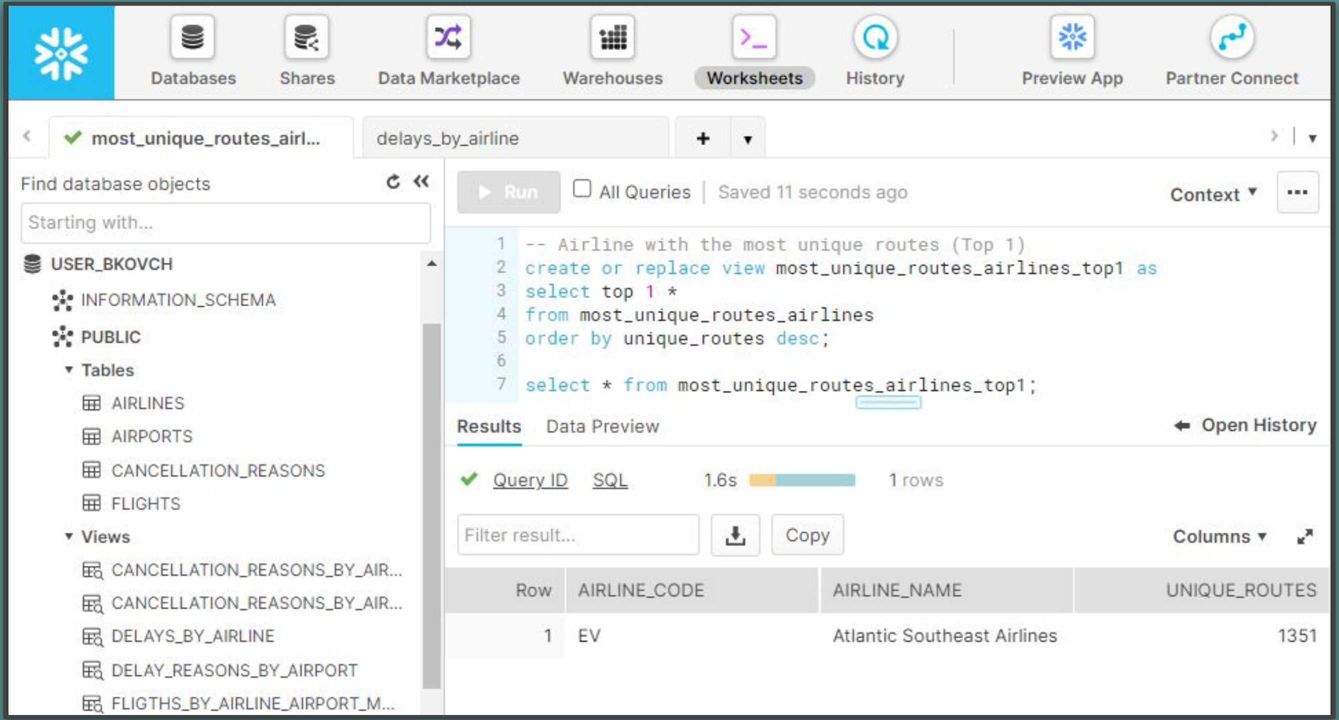


Read CSV Files

```python
def csv_to_df(file_location, file_type):
  # Provide parameters:https://community.cloud.databricks.com/?o=882228783267793#
  #   file location like "/FileStore/tables/phData_challenge/airlines.csv"
  #   file type like "csv"

  # CSV options
  infer_schema = "false"
  first_row_is_header = "true"
  delimiter = ","

  # The applied options are for CSV files. For other file types, these will be ignored.
  df = spark.read.format(file_type) \
    .option("inferSchema", infer_schema) \
    .option("header", first_row_is_header) \
    .option("sep", delimiter) \
    .load(file_location)

  return df

df_airlines = csv_to_df("/FileStore/tables/phData_challenge/airlines.csv", "csv")
df_airports = csv_to_df("/FileStore/tables/phData_challenge/airports.csv", "csv")
df_flights = csv_to_df("/FileStore/tables/phData_challenge/flights/", "csv")
```

Snowflake

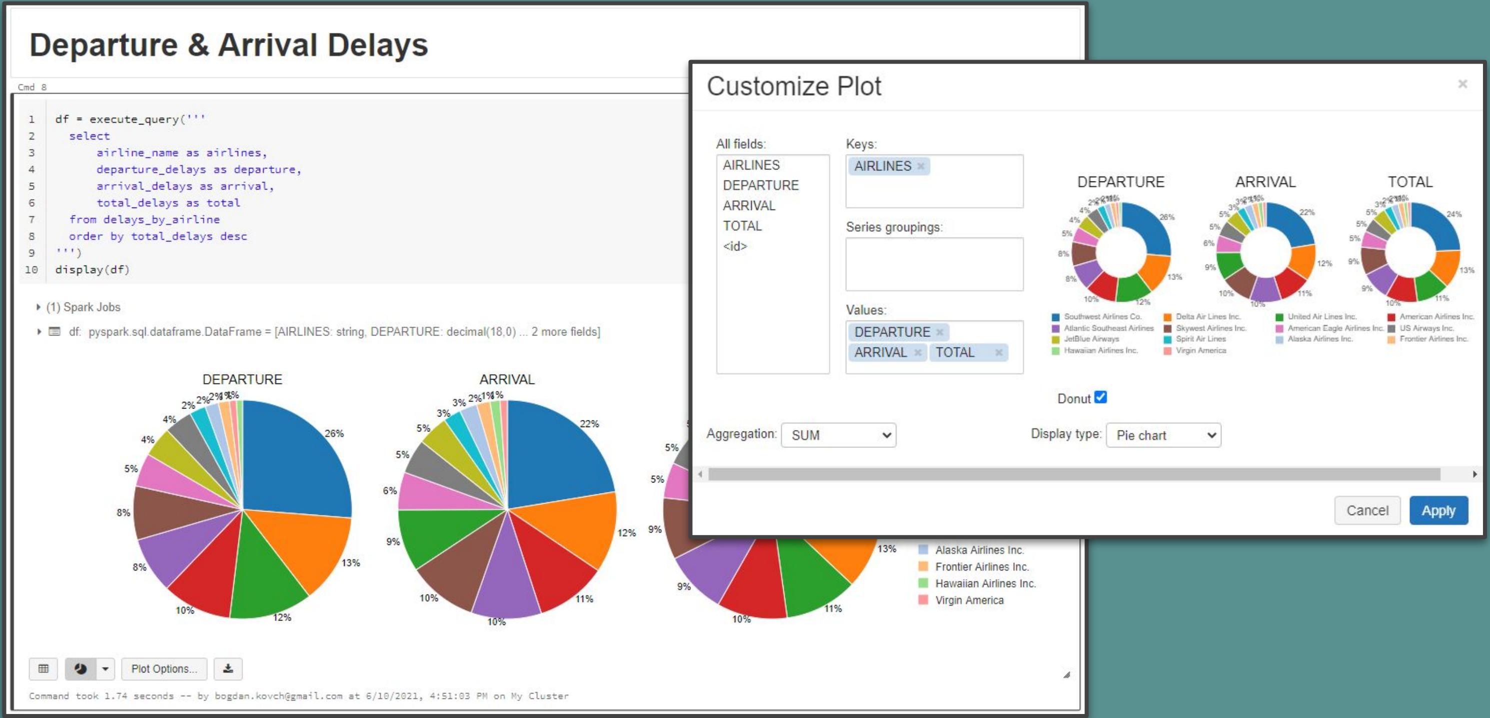
Data Warehouse



```sql
-- Airline with the most unique routes (Top 1)
create or replace view most_unique_routes_airlines_top1 as
select top 1 *
from most_unique_routes_airlines
order by unique_routes desc;

select * from most_unique_routes_airlines_top1;
```

Row	AIRLINE_CODE	AIRLINE_NAME	UNIQUE_ROUTES
1	EV	Atlantic Southeast Airlines	1351

Visualization in Databricks



Departure & Arrival Delays

Cmd 8

```
df = execute_query('''
  select
      airline_name as airlines,
      departure_delays as departure,
      arrival_delays as arrival,
      total_delays as total
  from delays_by_airline
  order by total_delays desc
''')
display(df)
```

▸ (1) Spark Jobs

▸ df: pyspark.sql.dataframe.DataFrame = [AIRLINES: string, DEPARTURE: decimal(18,0) ... 2 more fields]

```
Command took 1.74 seconds -- by bogdan.kovch@gmail.com at 6/10/2021, 4:51:03 PM on My Cluster
```

Customize Plot

All fields: AIRLINES, DEPARTURE, ARRIVAL, TOTAL, <id>

Keys: AIRLINES

Series groupings:

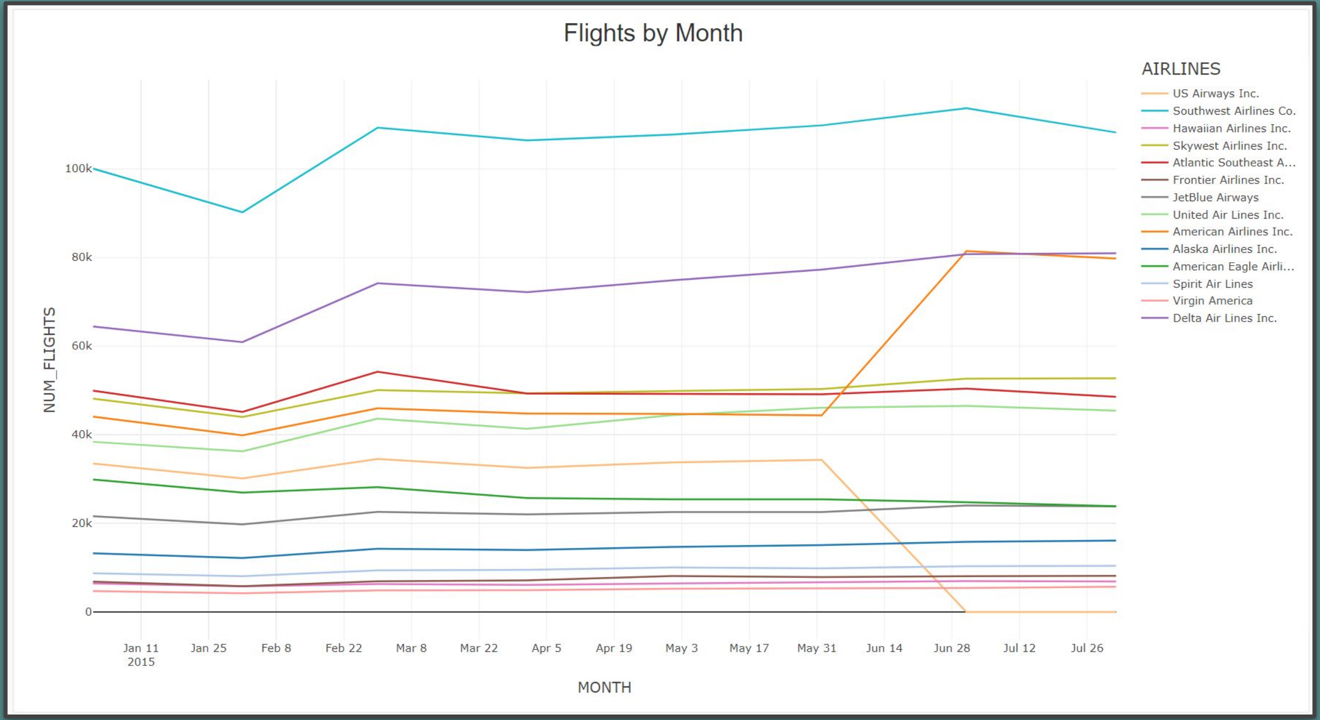
Values: DEPARTURE, ARRIVAL, TOTAL

Donut ☑

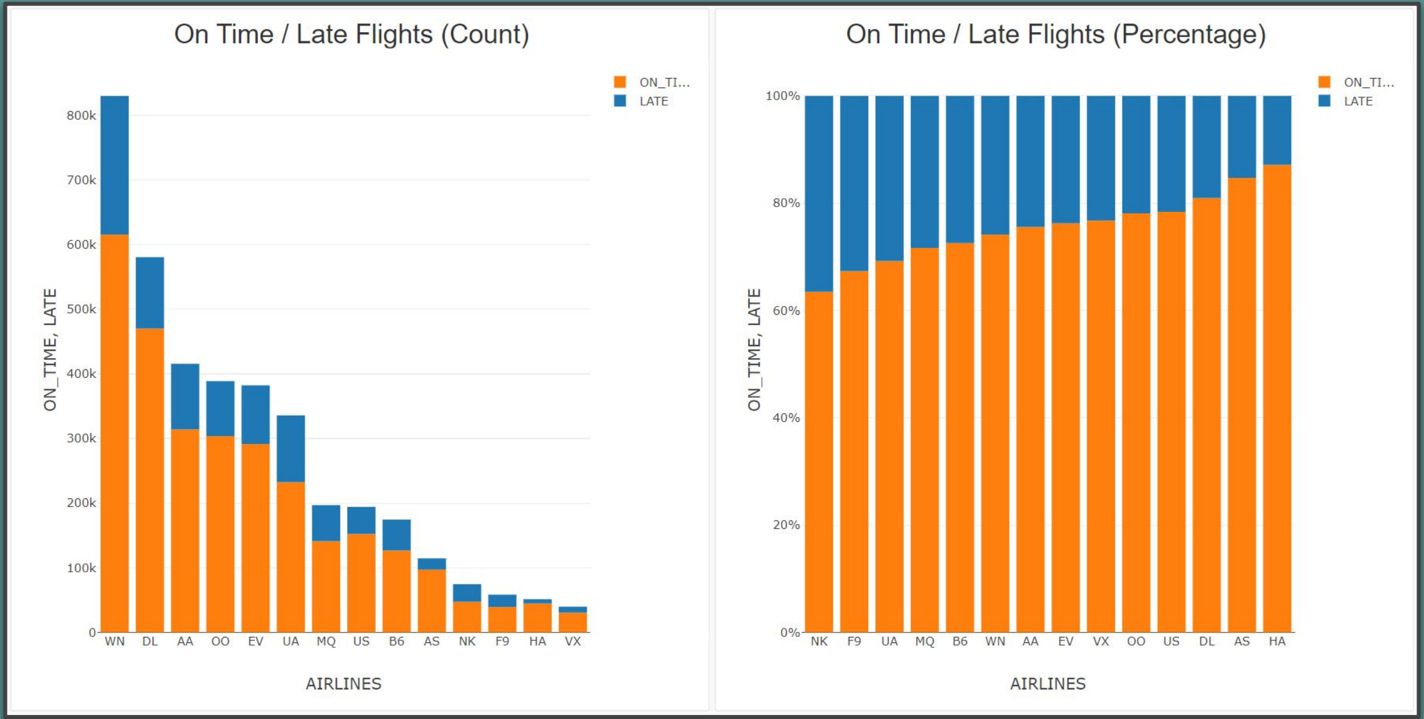
Aggregation: SUM

Display type: Pie chart

Monthly Flights by Airline



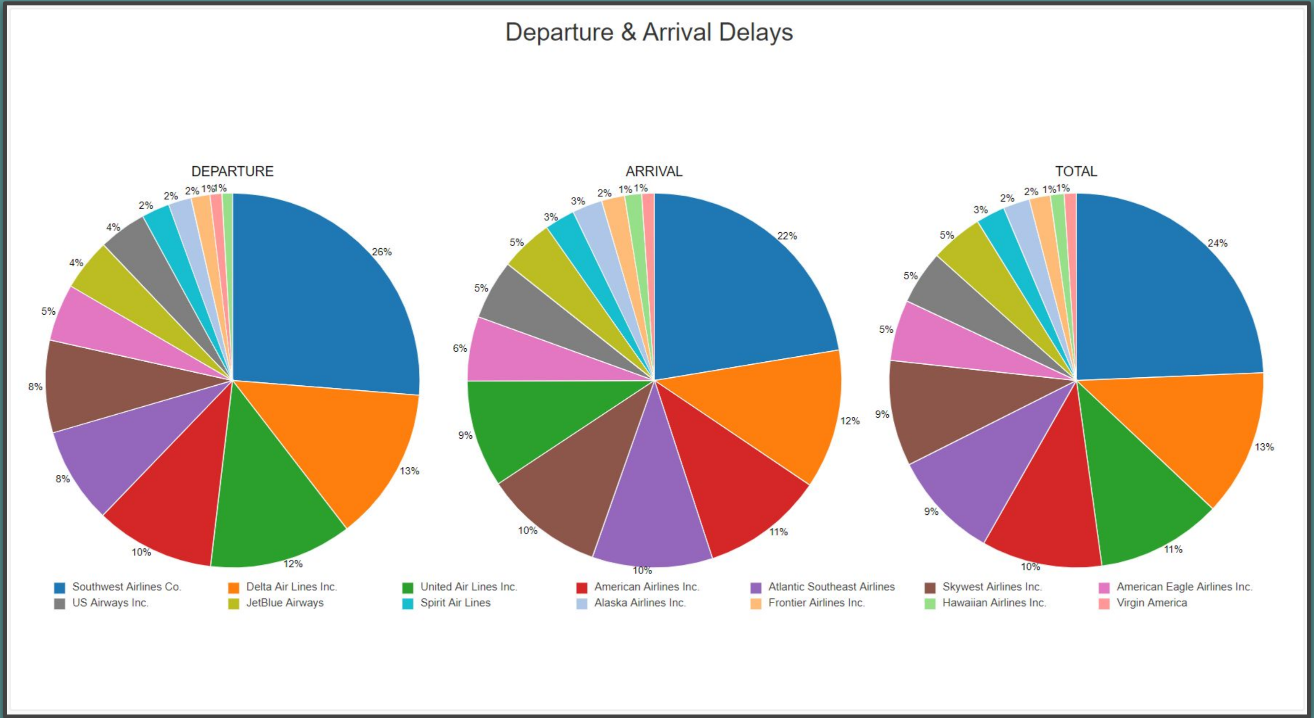
On Time vs. Late Flights



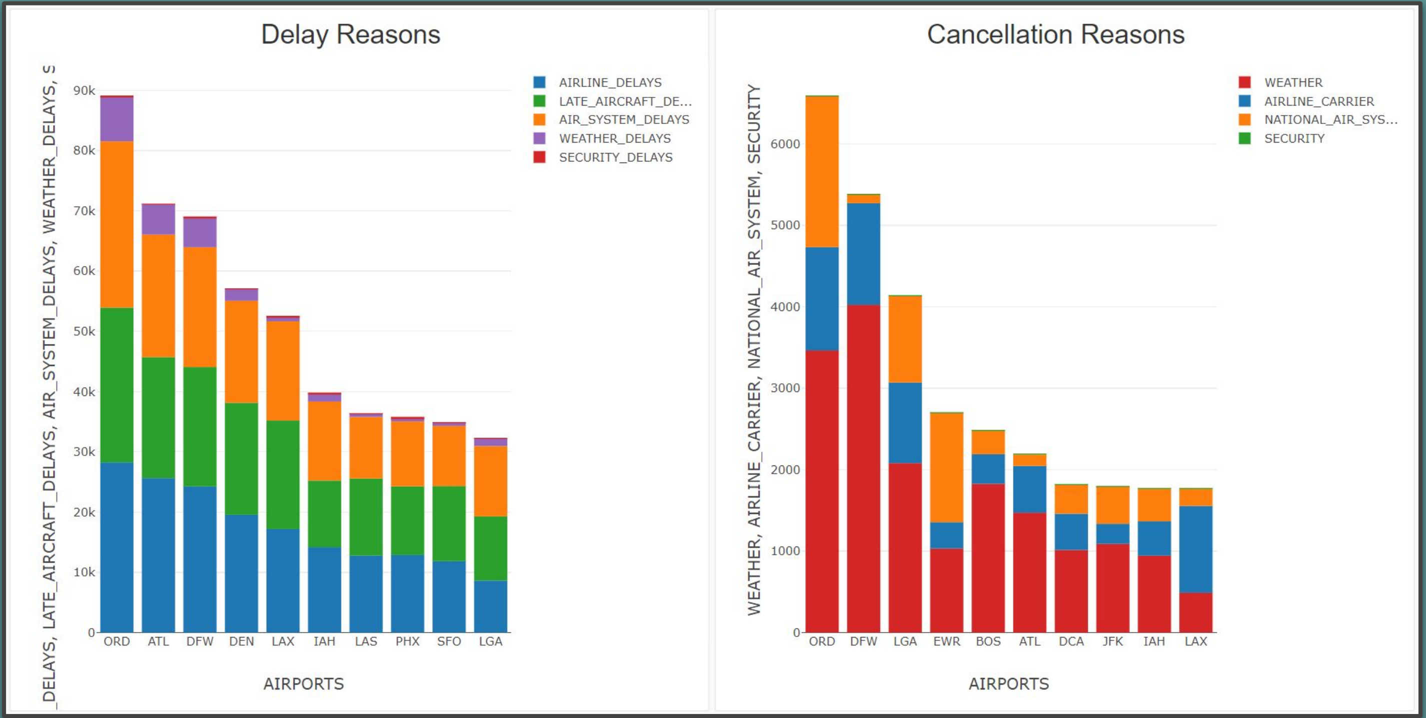
On Time / Late Flights (Count)

On Time / Late Flights (Percentage)

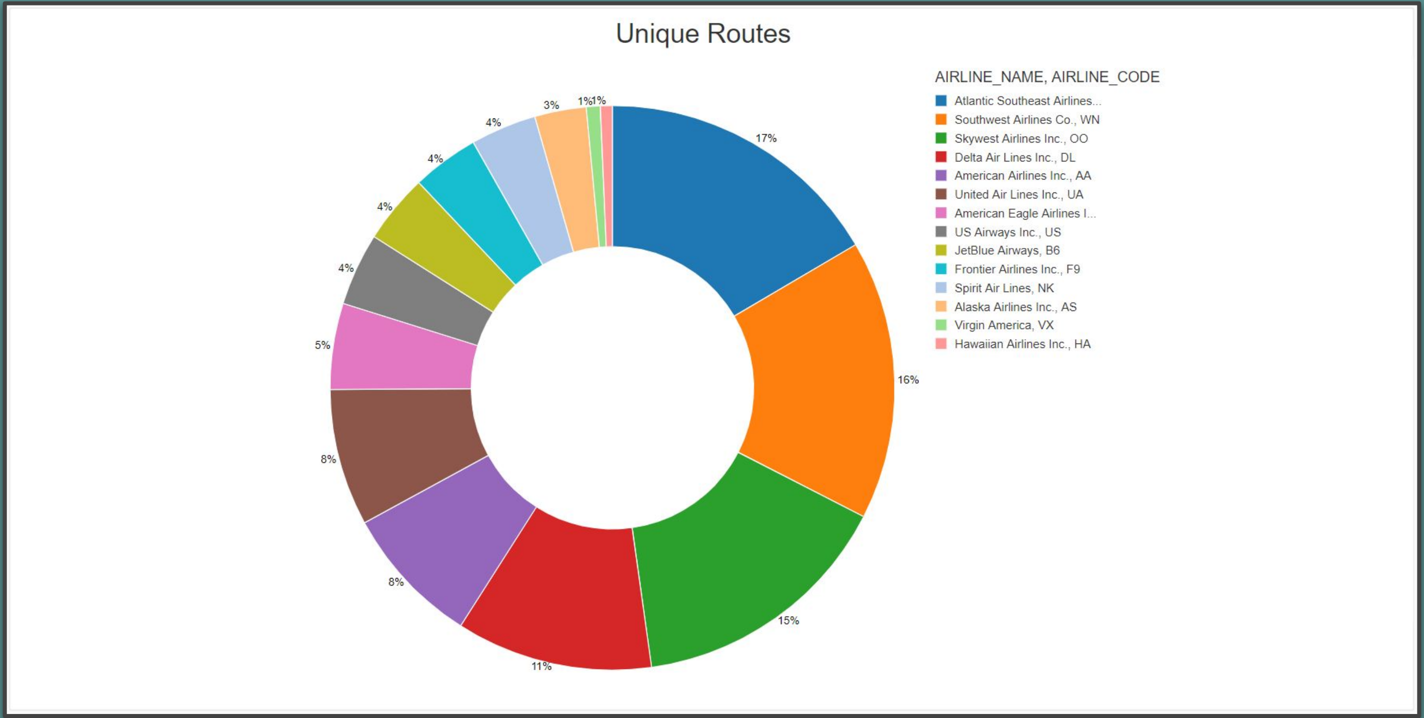
Departure & Arrival Delays



Delay & Cancellation Reasons



Variety of Routes



Prepared by Bogdan Kovch, June 2021
https://github.com/bkovch/DataChallenge
https://www.linkedin.com/in/bogdan-kovch/